## IN THE SPECIFICATION

The specification was objected to because of the following informalities:

- Section (g) which should address a brief summary of the invention regarding "identifying program phase changes through program working set analysis"
- Section (k) is a brief narrative of the disclosure as a whole in a single paragraph of 150 words or less.

Please amend the specification as follows:

Please insert the following beginning at line five of page two of the specification as originally filed, before the heading of "BRIEF DESCRIPTION OF THE FIGURES":

### SUMMARY

The disclosure presents systems and methods to identify program workings sets, detect working set changes and estimate working set sizes. The system generates a highly compressed representation of the working set, called a working set signature, by hashing working set elements into a data structure and setting the entries touched. The working set signature identifies, or is a representation of, the working set. The system can detect a working set change by comparing the signatures of consecutive working sets using a metric called a relative signature distance. The working set size is estimated by counting the number of bits set in the signature. The system can be used to compactly represent various types of working sets such as instruction, data and branch working sets. The system can detect program working set changes (or phase changes) independent of any micro-architectural specification. Thus, the system can be applied to any microprocessor without any modifications. Also, the system can be used to directly configure, i.e., without a trial and error process, certain hardware structures whose performance depends on the working set size. Such structures include caches and branch predictors. Also, the system can efficiently identify recurring program working sets using their associated signatures. The system can store signatures and associated optimal configurations for different working sets. When a working set repeats itself during program execution, the system can set the optimal configuration without going through a trial and error process. This can lead to significant reduction in time spent in non-optimal configurations.

Please amend the Abstract as provided on page 15, lines 3-22 of the application as follows:

The disclosure presents systems and methods to identify program workings sets, detect working set changes and estimate working set sizes. The system generates a highly compressed representation of the working set, called a working set signature, by hashing working set elements into a data structure and setting the entries touched. The working set signature identifies, or is a representation of, the working set. The system can detect a working set change by comparing the signatures of consecutive working sets using a metric called a relative signature distance. The working set size is estimated by counting the number of bits set in the signature. The system can be used to compactly represent various types of working sets such as instruction, data and branch working sets. The system can detect program working set changes (or phase changes) independent of any micro-architectural specification. ~~Thus, the system can be applied to any microprocessor without any modifications. Also, the system can be used to directly configure, i.e., without a trial and error process, certain hardware structures whose performance depends on the working set size. Such structures include caches and branch predictors. Also, the system can efficiently identify recurring program working sets using their associated signatures. The system can store signatures and associated optimal configurations for different working sets. When a working set repeats itself during program execution, the system can set the optimal configuration without going through a trial and error process. This can lead to significant reduction in time spent in non-optimal configurations.~~